Towards automated protest event analysis

Makarov, Peter ; Lorenzini, Jasmine ; Rothenhäusler, Klaus ; Wüest, Bruno

Draft: Towards Automated Protest Event Analysis

Peter Makarov[1], Jasmine Lorenzini[2], Klaus Rothenhäusler[1], and Bruno Wüest[3]

[1]*Institute of Computational Linguistics, University of Zurich*
[2]*European University Institute, Florence*
[1]*Department of Political Science, University of Zurich*

28th June 2015

1. Introduction

Protest event analysis (PEA) is an important method of social movement research (Hutter 2014). A type of content analysis, it studies public protest on the basis of news documents. A protest event is open to the public, politically motivated and not institutionalised as opposed to e.g. elections. In traditional PEA, a human annotator identifies a protest event in a document and characterises it in terms of a small set of attributes such as action form, issue, actors, location and time, as Figure 1 illustrates.

Trade unions are satisfied with the course of today's blockade of the Czech-Slovak Drietoma-Stary Hrozenkov border crossing aimed to highlight bad social and economic conditions in Slovakia ...

Czech News Agency, 2 March 2001

action form	blockade
actor	trade unions
issue	welfare
location	Slovakia
date	02.03.2001
...	...

Figure 1: PEA: From document to structured representation of information on a protest event

Many documents need to be processed this way before any statistical analysis can be performed on the output data. To reduce the amount of manual annotation, natural language processing (NLP) technology has been applied to the problem (Hanna 2014; Leetaru 2011).

2. Document classification

It is readily seen that PEA is an instance of information extraction (Piskorski and Yangarber 2013), a major NLP task of deriving structured information from unstructured textual data. In practice, PEA is a two-step procedure. First, to obtain a relevant document collection, one typically issues a very general query to a data service like LexisNexis[1] in the hope of retrieving almost all relevant documents (Schrodt 2010). The downside of this is that the number of false positives is high. If a project is ambitious, one easily ends up having to filter through millions of documents. Thus, the first objective dictated by practical considerations is document classification and de-duplication. Information extraction only comes as a second step.

To tackle the document classification task, one summons the classic technique of supervised statistical document classification – fitting a statistical classifier to a set of labelled documents modelled with a bag-of-words document model (Sebastiani 2002). This technique is known to perform very

[1] `http://www.lexisnexis.com`

well on news documents (e.g. the classic multi-label benchmark Reuters-21578, ModApté split[2]), even in skewed problems when the number of instances of the positive class is much lower than that of the negative class (Dumais *et al.* 1998). Often a skewed problem with positive rates reported as low as 0.5% (Hanna 2014), document classification for protest and related domains achieves modest success even when the amount of labelled data is large. For example, Leetaru 2011 describes a chain of two classifiers trained on 30,000 documents that retrieves virtually all protest-relevant documents, but its precision ranges from 50% to 66% depending on the news source[3]. This is one of the top results reported for the problem.

For the needs of the POLCON project[4] of the European University Institute and the 'Years of Turmoil' project[5] of Zurich University, we have produced a data set of 7,033 labelled news documents that we have retrieved from LexisNexis[6]. The documents come from major transnational news agencies like Agence France-Presse and the Associated Press, and national agencies like the Czech, Polish and German News Agencies. The data set features a total of 12 news sources and spans the period from 2000 to 2014. The rate of relevant documents is slightly above 10% and is higher than the rate in the population, which we estimate at around 5%. This is due to the fact that from a certain point on, we have used a classifier to pre-select documents. (Inter-annotator agreement figures.) We present some statistics for the data set in the appendix.

For the document classification task, we have experimented with a number of standard feature set-up choices like the application of stemming, various n-gram ranges, generation of features from all sentences vs only those containing strong protest keywords, upweighting such sentences etc. Likewise, our choice of classifier is standard and guided by common recommendations for document classification (e.g. Manning *et al.* 2008, section 14.6). We typically fit a linear classifier using the `sklearn`[7] implementation of the stochastic gradient descent algorithm. We regularise it with the elastic net (Zou and Hastie 2005) that also does feature selection for us. We perform cross validation for hyper-parameters with respect to F_β-score, $\beta > 1$ that puts more weight on recall. In the end, our classifiers perform in the 40-45% range for precision and 75-80% for recall.

Our approach suffers from high variance in the data. One observation is that classifier performance as a function of the training set size plateaus after about 2,000 documents, as Figure 2 demonstrates. Our checks suggest that this is not an artefact of the test set.

[2] `http://www.daviddlewis.com/resources/testcollections/reuters21578/`

[3] Here and elsewhere, recall, precision, and F_β-score are for the positive class unless stated otherwise.

[4] POLCON – Political conflict in Europe in the shadow of the great recession: `http://www.eui.eu/Projects/POLCON`

[5] 'Years of Turmoil' – Political consequences of the financial and economic crisis in Europe: `http://www.mwpweb.eu/SiljaHaeusermann/research_current_projects.html`

[6] The following query was issued: `initiative OR referendum OR petition! OR signature! OR campaign! OR protest! OR demonstrat! OR manifest! OR marche! OR marchi! OR parade OR rall! OR picket! OR (human chain) OR riot! OR affray OR festival OR ceremony OR (street theatre) OR (road show) OR vigil OR strike! OR boycott! OR block! OR sit-in OR squat! OR mutin! OR bomb! OR firebomb! OR molotov OR graffiti OR assault OR attack OR arson OR incendiar! OR (fire I/1 raising) OR (set AND ablaze) OR landmine OR sabot! OR hostage! OR assassinat! OR shot OR murdered OR killed`

[7] `http://scikit-learn.org`

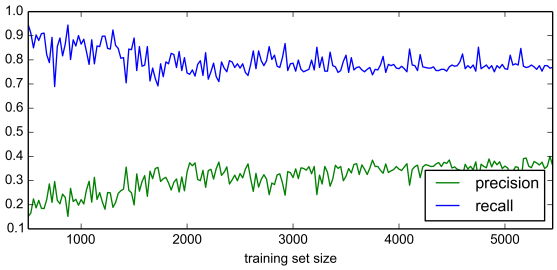


Figure 2: Test set precision and recall as functions of training set size: linear SVM + elastic net penalty, 3-fold CV w.r.t. F_2. Results averaged over 4 runs, each with a random permutation of the training set.

On the one hand, our data set is clearly sparse. There are many idiosyncrasies in the relevant documents due to the differences in the location and time of protest events as well as protest forms that range from strikes and demonstrations to violent protest, to subtle symbolic protest actions. (Performance figures by protest form). To address sparsity, we applied some semi- and unsupervised techniques that exploit the structure of the unlabelled data. One approach uses topics of a topic model (Blei *et al.* 2003) as features in a document classifier. We have no conclusive results yet, however experiments on a different data set from the same domain have produced significant improvements (Wueest *et al.* 2013).

The other aspect of the problem is that the features that we engineer, following the standard recipe for document classification, fail to capture some important distinctions. We believe that the bag-of-words document model loses us important information about the local context in which protest keywords appear.

This and the fact that we wanted to make some quick progress with the information extraction task have brought us to consider more closely the operational definition of protest that we have used.

3. Protest

Common expressions for protest forms are many-way ambiguous, which is why they make poor keywords. WordNet (Miller 1995) lists 21 synsets (senses) for the verb *strike* alone. Some senses are more likely to occur in a news document than others. Domain ambiguity – e.g. *strike* as protest vs *strike* as hit in sports – typically resolves well at the document level, as we also observe in our results. Other kinds of ambiguity are strictly local: To identify the sense one needs to consider the immediate context or the sentence the word occurs in. Whereas simple techniques like n-grams and k-skip-n-grams could be suggested as a means to help a document classifier disambiguate in some cases (e.g. for set phrases like *strike an agreement*), things get quickly out of hand when fine-grained distinctions need to be taken into account (citation from word-sense disambiguation literature).

PEA adds even more ambiguity by introducing additional constraints as to what exactly counts as a relevant protest mention. Prepared under rigorous guidelines, our document collection is a good example of what a common data set for PEA is like. The NLP side of the project have provided little input on what should be considered as a relevant document. Error analysis has shown that certain aspects of the operational definition of protest event, under which the annotator judges on document relevance, are particularly bad sources of classification error (Figure: breakdown by error). None of them are adequately addressed in the features.

i) Historical events and commemorations: The same challenge is reported by Schrodt 2010.

ii) Location of a protest event: The POLCON project studies protest activity in 27 European Union member states (excluding Croatia that joined only in 2013), Iceland, Norway, and Switzerland. We could generally improve on filtering out irrelevant locations with two simple tricks: Either with the help of the meta-data on story locations that LexisNexis provides or by throwing in location relevance as another feature, with the help of a small manually compiled gazetteer.

iii) Factuality: A protest event has to be presented in the text as factual – either as having taken place or planned with a date and location clearly stated. In particular, protest threats, except bomb threat, and generic mentions of protest do not count as relevant.

Other projects dealing with PEA are likely to face these problems as well and a hard decision as to how to go about them. At this point, it has become clear that in order to advance on document classification, we have to be able to identify potential protest mentions in a document and verify whether they satisfy the definition.

4. Anchors

For NLP practitioners, PEA is a classic event extraction (EE) problem in the sense of the Knowledge Base Population track of the Test Analysis Conference[8]. EE specifically seeks to identify in the text *who did what and where, when, how, due to what* and so on. One also has to infer whether an event has actually occurred (e.g. ACE 2005). EE is a difficult task even for human annotators, which explains the relatively low performance of EE systems (Piskorski and Yangarber 2013; Yangarber and Grishman 1997).

One would typically break down an EE problem like PEA into sub-tasks and first attempt to solve them in isolation. Some commonly identified sub-tasks of document-level EE are anchor expression identification, attribute identification[9], and event co-reference resolution (Ahn 2006). Here, an anchor (or a trigger) is a term that expresses the occurrence of an event most clearly. For example, *demonstrated* is an event anchor in the sentence *Teachers demonstrated in front of the local ministry of education.* Attributes of an event are its location, time, actors, issue, factuality, etc. One strategy for EE would be to first identify anchors, then explore their local context for information on attributes, and finally perform information consolidation across event mentions.

We have started our exploration of PEA as an EE task by attempting to tackle the problem of identification of protest event anchors. We have tried out two sentence-level approaches: pattern matching with patterns bootstrapped from unlabelled data, and applying a supervised statistical anchor classifier trained on a standard set of features from tasks like named entity (NE) recognition and relation extraction (Ahn 2006; Ratinov and Roth 2009).

Our teams have manually annotated relevant documents at the token level for mentions of protest events and their attributes: actors, location, time, number of participants, issue, stance on the issue. We have used browser-based annotation tool `brat`[10] (Stenetorp *et al.* 2012). The task was to identify spans of text that most clearly and concisely express events and attributes, classify them into sub-types, indicate within a sentence which attribute belongs to which event, and index co-referring event mentions (Figure 3).

[8]The successor of the Message Understanding Conference and Automatic Content Extraction Programme, the Test Analysis Conference has been hosting since 2009 the most important evaluations in the field of information extraction: `http://www.nist.gov/tac/`

[9]For simplicity, we will refer to both arguments and attributes in terms of ACE 2005 and Ahn 2006 as attributes.

[10]`http://brat.nlplab.org/`

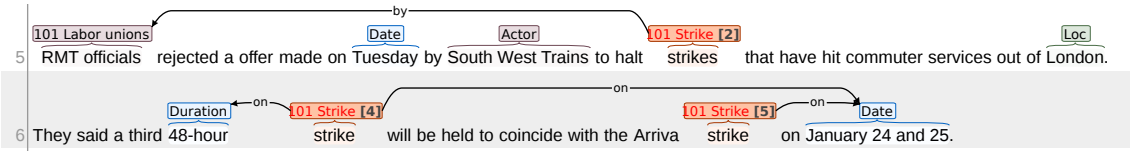


Figure 3: A sample of annotations in `brat`: Locations, dates, and actors come partially pre-annotated with Stanford NE recogniser (Finkel *et al.* 2005). The annotator indexes co-referring protest event mentions.

The task has proven difficult. We have used an in-house metric to compute inter-annotator agreement. For a pair of annotators, the metric first tries to find the optimal match between two sets of events. An event is represented by an action form type and a set of word types from all co-referring text spans. We rate matches between events by computing the overlap coefficient between two word type sets weighted by a non-zero event form type match. The metric rewards the identification of the same words irrespective of their exact position in the text and is lenient towards action form type mismatch. At this step, the metric does not take account of attributes, which is too rigid in cases when an anchor is too difficult to pin down. Matches between attributes of two matched events are computed in the same way. We then compute pairwise F-scores (Hripcsak and Rothschild 2005) and average them on the document or event level. So far, we have measured inter-annotator agreement only once between two rounds of annotation. The numbers shown in Table 1 should be considered as development-stage. Subsequently, substantial changes have been made to the operational definition of protest event and the overall annotation procedure. At the end, the annotations have all been double-checked.

	document level	event level
μ	0.484	0.527
σ	0.113	0.096

Table 1: Development-stage inter-annotator agreement measured by average pairwise F-score.

So far, in our work we have only looked at the annotations of event mentions. We should mention that some aspects of the annotations, although contributing to the low scores, have nothing to do with the labelling of event mentions. For example, our definition of protest requires that an event have at most one location and time attribute. Thus, if a sentence reads *strikes in London, Manchester, and York*, the annotator has to annotate three events. In practice, this rule is quite difficult to apply, yet the metric would penalise harshly for any unlabelled event.

5. Statistical anchor classifier

We run the documents annotated in in this way through the Standford CoreNLP pipeline (Manning *et al.* 2014), to obtain dependency parses, lemmas, part-of-speech (POS) and NE tags. We have used dependency parses to infer the set of anchors. An anchor is any notional word from a span labelled as an event whose dependency head lies outside the span. There are few surprises among most frequent anchors. One hundred most frequent anchors account for over 74% of all event mention annotations and about 73% of anchors occur only twice or less.

Although this definition is very convenient, it is not the optimal way to pick anchors. One option would be to always try to select the strongest keyword of the span, so that, e.g. *protest* is selected over *stage* in event mention *stage a protest*. Alternatively, one could cast the task as a sequence classification problem (Lafferty *et al.* 2001). We have found that except for words *protest* (31%) and *attack* (14%), it is rare that a strong keyword is not an anchor of the event mention that it occurs in (around 6% of the cases).

Our goal is to predict whether a given word is the anchor of a protest event. We have taken the standard approach and have been developing a statistical classifier that learns from the event

lemma	POS	count	lemma	POS	count	lemma	POS	count	lemma	POS	count
strike	N	261	gather	V	38	suspend	V	1	SLOGAN	N	1
protest	N	202	bomb	N	36	arrest	V	1	condemn	V	1
demonstration	N	132	action	N	35	hunt	N	1	MEETING	N	1
attack	N	125	attack	V	34	barricade	N	1	forum	N	1
rally	N	67	clash	N	33	information	N	1	arrive	V	1
protest	V	48	blockade	N	32	submit	V	1	struggle	N	1
march	N	47	march	V	30	toss	V	1	demolish	V	1
block	V	42	letter	N	28	draw	V	1	BOMBING	N	1
demonstrate	V	40	petition	N	26	Kurds	N	1	prompt	V	1
riot	N	39	throw	V	24	initiative	N	1	bundle	N	1

Table 2: Twenty most frequent and some least frequent anchors.

mention annotations. So far, we have conducted preliminary checks with features up to the sentence level. We add path features in the hope of better capturing local information about attributes. Our features are inspired by what is suggested in the literature on anchor identification (Ahn 2006) and common token-level information extraction tasks.

We generate the following features for each verb or noun whose NE tag is a blank. For now, we exclude other lexical categories since we find that common nouns and verbs make up more than 95% of all anchors.

i) The lemma, POS tag, stem by the English Snowball stemmer (Porter 1980, 2001) of the word. If the word is a noun, we check in WordNet whether it is animate.

ii) Similarly for two notional words to the left and two to the right. Additionally, we take their NE tags. In case a word is a location or MISC(ellaneous), we look it up in our small gazetteer of relevant and irrelevant locations or adjectives like *British, Irish*, and check a feature: relevant/irrelevant/unknown.

iii) We take at most two shortest dependency paths of up to a certain length towards animate nouns, locations and MISC(ellaneous) terms. For example, for *protest* from *a protest organised by environmental activists*, we would generate a path feature $\xrightarrow{vmod}$ *organise* $\xrightarrow{agent}$ that points to animate noun *activists*. We also add relevance features for the words the paths lead to. So far, our solution to incorporating time information has been rudimentary: We map relevant (2000 on) and irrelevant years to different strings and use them instead of the lemmas.

iv) We add as features the set-of-lemmas of the notional words in the sentence.

v) We use bootstrapped or otherwise constructed patterns that match protest events (e.g. *take to streets, shout slogans*) and a list of bootstrapped protest actor terms (e.g. *demonstrator, supporter, activist, rioter*). In this way, we try to incorporate some external knowledge about protest. If the word is an animate noun, we check if it is on the actor term list and add this as a feature. For verbs and nouns, we try to match any of the patterns that contain them, within a certain window around the word. We also add as a feature the number of patterns the word appears in.

	lemma	POS	stem	NE	anim.	rel.	pattern match	pattern frequency
word −2	catholic	JJ	cathol	MISC	0	0		
word −1	teen-ager	NN	teen-ag	O	0	0		
word	shoot	VBN	shoot		0		0	2
word +1	kill	VBN	kill	O	0	0		
word +2	earlier	RBR	earlier	DATE	0	0		

	path	lemma	POS	NE	actor	rel.
actor path 1	$\xrightarrow{conj\_and}$ kill $\xrightarrow{prep\_by}$ paramilitary $\xrightarrow{nn}$	Protestant	(NNP)	MISC	0	0
location path 1	$\xleftarrow{ccomp}$ ominous $\xrightarrow{conj\_and}$ tear $\xrightarrow{nsubjpass}$	Belfast				1
misc path 1	$\xrightarrow{nsubjpass}$ teen-ager $\xrightarrow{amod}$	catholic	JJ			0

Table 3: Some of the features for *shot* from the sentence *Other portents were ominous, however: Belfast was torn late last month by the worst sectarian riots in three years, a Catholic teen-ager was shot and killed earlier this month by suspected Protestant paramilitaries...* Upper table: features contributed by *shot* and its neighbouring words, and features related to pattern-matching (is there a match by a pattern? How many patterns contain *shot*?). Bottom table: path features, only one path per category found.

We have trained a linear classifier using stochastic gradient descent in much the same way as for the document classification task except that we have cross-validated with respect to F-score. We have used 340 documents for training. We present an evaluation for the development subset of the set of the relevant documents (78 documents). For the baseline, we have generated all 2-skip-n-grams, $n \in \{1, 2\}$, of every event mention annotation from the training set and all 3-skip-n-grams, $n \in \{2, 3\}$, from the window of 4 notional words around every event mention with less than 3 notional words, such that the skip-gram contains at least one word from the event mention. For example, if *strike* is annotated in *stage a nationwide strike to demand a pay rise*, then we create skip-grams (*stage, strike*), (*nationwide, strike, demand*), etc. We use skip-grams as regular expressions to match sentences that contain event mentions. A skip-gram matches a sentence if every token from a skip-gram occurs in the sentence and there are at most 4 notional words between adjacent tokens. We have computed the precision of each skip-gram with which it identifies sentences with event mentions, using the sentences from the training set minus the sentence the skip-gram originates from. We have used these precision scores to filter out noisy skip-grams. The baseline results are for the combination of event-mention-only skip-grams with precision ≥ 0.56, context-based skip-grams with precision ≥ 0.8, and all skip-grams unseen in training. Table 4 compares the results.

	precision	recall	F-score	positive rate
baseline, sentence level	0.57	0.59	0.58	228/1,616
anchor classifier, token level	0.49	0.46	0.47	252/10,463
anchor classifier, sentence level	0.64	0.53	0.58	228/1,616

Table 4: Preliminary evaluation of performance of the statistical anchor classifier.

(Breakdown by feature groups.) Error analysis suggests that the the information available up to the sentence level is often not enough to decide on the label. Further, threats and other non-factual uses of protest terms pose a problem to the classifier and features that would specifically address this are needed. We expect the performance to go up once we fix these and other feature set deficiencies, e.g. a better choice of anchors, a better source of information on location relevance.

We will now turn to the bootstrapping of protest event patterns – another approach to anchor identification that we have explored.

6. Bootstrapping

Traditionally, pattern matching has been a core method of information extraction and is still in active use (Yangarber and Grishman 1997; Du and Yangarber 2015). In pattern matching, one applies extraction patterns to the document in the hope of locating relevant information. Extraction patterns are simply regular expressions that classify terms that they match. Patterns are typically high-precision extractors. For example, *protest organised by* X_{NP} could be a pattern that matches whatever term occurs in the position of the noun phrase X_{NP}, e.g. *trade unions*, and classifies it as a protest actor.

Ideally, one wants to automate pattern acquisition as much as possible. One influential idea has been bootstrapping – a family of iterative heuristic-based semi-supervised algorithms (for an overview, see e.g. McIntosh 2009, chapter 4). The naive assumption behind bootstrapping is that if a set of terms of a relevant category all co-occur with a pattern, then this pattern is likely to be a good extractor for terms of this category and so new terms that this pattern extracts are likely to be members of that category. In this way, one iteratively builds collections of terms and patterns. Although the naive assumption clearly does not always hold, with proper constraints, it works well for language data (Abney 2007).

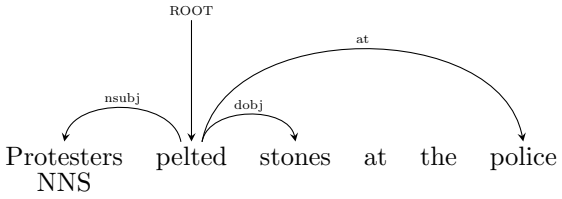
One relevant approach (Huang and Riloff 2013) uses bootstrapped patterns to identify sentences that speak about civil unrest. The idea is to learn dictionaries of patterns from sentences of the form *actor–event–purpose*, e.g. *Workers took to the streets to demand better working conditions*. Candidate patterns are generated from dependency parses produced by the Stanford parser (Klein and Manning 2003; De Marneffe *et al.* 2006) on the English Gigaword corpus (Parker *et al.* 2011). Actor terms are taken to be subject nouns (*workers, protesters, demonstrators*), event phrases are main verbs with a complement or prepositional adjunct (*took to streets, threw stones*), and purpose phases are non-finite verbal adjuncts (*to demand conditions, protesting against war*). At each iteration of bootstrapping, two types of entities promote candidates of the third type to the new round. For example, if a sentence contains both a relevant event phrase and a relevant purpose phrase, then its actor term is added to actor terms for the next iteration. The detection of event mentions in the text is performed via dictionary look-up. Huang and Riloff apply bootstrapped patterns to the document classification problem. If patterns of at least two types match in the same sentence, the document gets labelled as relevant. Their approach outperforms a set-of-words document classifier.

The encouraging results and simplicity of their approach have lead us to experiment with bootstrapped patterns. Our bootstrapping builds on their work. We explore co-occurrences of plural nouns like *protesters, workers, demonstrators* and predicates of which they are agents: *occupy square, burn flag, block road*, etc. We use dependency parsing to generate candidates. We expand the set of sentences that could contribute candidates to include passive constructions, relative clauses, and more types of verbal adjuncts. However, we consider purpose phrases a subset of event phrases, partly because we observe that the same predicate could occur as both. Additionally, the original proposal tries to identify purpose phrases, like in the example sentence above, with the help of the *xcomp* relation, which is in fact a parser error.

For our experiments, we have taken 1.8 million documents totalling 812 million tokens. These are all unique documents from AFP, DPA, BBC and PRS from the years 2000 to 2014 that we could retrieve from LexisNexis with our query of protest keywords. We have run the documents through the Standford CoreNLP pipeline to obtain dependency parses, lemmas, part-of-speech (POS) and named entity (NE) tags. To identify candidates, we have used collapsed dependencies with propagation of conjunct dependencies (De Marneffe *et al.* 2006). A term candidate is a lemma. A pattern candidate is a triple of lemmas of a verb and its (direct or prepositional) object

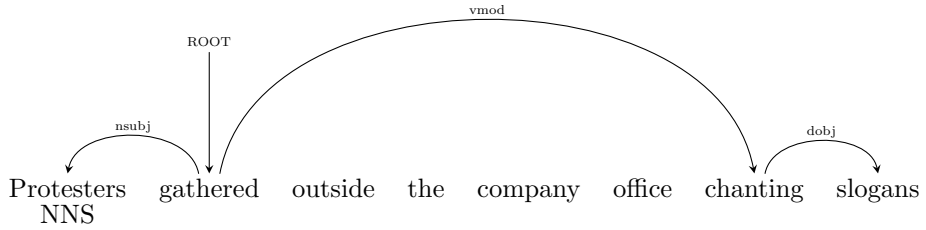
or prepositional adjunct (i.e. a dependent in any relation whose label matches `dobj|prep_.*`), and the preposition if any. However, if the dependent is a NE, we store the NE tag and discard the lemma. If a sentence matches one or more of the following cases, we extract new candidates and update statistics of co-occurrence of term and pattern candidates:

i) The sentence has a plural common noun subject. In this case, the subject produces a term candidate and the main verb produces a pattern candidate with each of its objects and prepositional adjuncts.



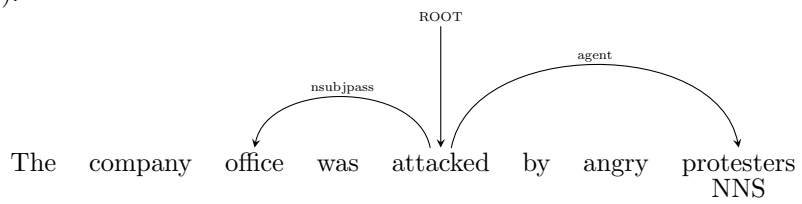
From this sentence, we harvest a term candidate *protester* and pattern candidates (*pelt, –, stone*) and (*pelt, at, police*).

ii) The sentence contains a plural common noun subject and the main verb has a dependent in the *xcomp* or *vmod* relation, i.e. a non-finite verbal complement or adjunct. In this case, the subject produces a term candidate and the non-finite verb produces a pattern candidate with each of its objects and prepositional adjuncts.



From this sentence, we harvest a term candidate *protester* and a pattern candidate (*chant, –, slogan*).

iii) The sentence contains an *agent* dependency relation whose dependent is a plural common noun. In this case, this noun produces a term candidate. The verb that governs it produces a pattern candidate together with its *nsubjpass* dependent, or *rcmod* or *vmod* head. This accounts for passives (*nsubjpass*) and finite and non-finite relative clauses (*rcmod* and *vmod* respectively).



From this sentence, we harvest a term candidate *protester* and a pattern candidate (*attack, –, office*).

Instead of the algorithm of Huang and Riloff, we have chosen a version of weighted mutual exclusion bootstrapping (McIntosh and Curran 2008). To address the problem of semantic drift – a situation when new terms and patterns move too far away from the initial relevant set – this algorithm learns multiple pairwise disjoint categories that constrain each other and makes sure old and new terms

and patterns contribute to bootstrap equally. Terms that co-occur with patterns of multiple categories are discarded, and equivalently for patterns. Thus, a common scenario would be to learn one target category alongside a number of stop categories (by analogy with stopwords). To rank candidate terms and patterns, the authors propose to use co-occurrence frequencies and break ties with the χ^2 statistic. (figure: algorithm)

We have excluded all terms and patterns that occur less that 3 times. As a result, we have got 589,856 unique patterns and 15,156 unique terms. We have tried the original ranking metric from the paper, but found the results produced by the classic *RlogF* metric in combination with χ^2 on ties better. For patterns, *RlogF* (Riloff and Jones 1999) is defined as:

$$RlogF(pattern_i) := R_i \log_2(F_i),$$

where F_i is the number of relevant unique terms that $pattern_i$ co-occurs with, N_i is the number of all unique terms that $pattern_i$ co-occurs with, and $R_i = \frac{F_i}{N_i}$. *RlogF* for terms is defined in the same way, with patterns instead of terms.

After some experiments, we have also decided to remove some numerals and group terms like *thousand* and *hundred* that add noise. We have also found that results improve if at each iteration, we learn more patterns than terms in a kind of meta-bootstrap fashion (Riloff and Jones 1999). For the results that we present, we have run bootstrapping for 415 iterations and at each iteration, the algorithm selected one new term and 12 new patterns. We have learned two categories: protest and war introduced as a stop category. We have seeded them with term and pattern seeds shown in Table 5.

Protest	seed terms	student, demonstrator, protester, protestor, activist, supporter, worker
	seed patterns	take to street, hold protest, chant slogan, go on strike, clash with police, walk off job
War	seed terms	troops, gunman, soldier, militant, fighter
	seed patterns	kill civilian, ambush soldier, fire rocket, fire missile
	filtered terms	thousand, hundred, group, ten, score, dozen, people, member, man, million, handful

Table 5: Seeds and filter terms

Table 6 presents some of the results. Generally, bootstrapped patterns look good however we have found that unacceptably many high frequency patterns like *block road* have got filtered out due to the stop category.

top patterns	march in anger, march in front of embassy, march towards station, burn in demonstration, vow despite ban, march with placard, shout outside building, march from palace, chained in LOCATION, march on legislature, scuffle on DATE, march from town, paint graffitus, throw during demonstration, chant during DATE, rally at square, break through security, chant after prayer, invade runway
top terms	longshoreman, tradesman, activitist, indian, hauler, steelworker, stevedore, wholesaler, hostess, anti-globalist, romanian, argentine, dockworker, metalworker, intern, schoolteacher, sympathiser, pharmacist, residence, steward, taxi, subcontractor, sympathizer, anarchist, detractor, vandal, counter-demonstrator, rioter, labourer

Table 6: Some bootstrapped patterns and terms

For the baseline, we have computed mutual information, commonly used in feature selection (e.g. Manning *et al.* 2008, section 13.5.1), between all the patterns and two classes of terms: the

protest seed terms from the table and all the rest. We have ranked the patterns by the mutual information score and selected top n patterns that co-occur with the protest seed terms. Top n bootstrapped patterns are picked with respect to the order in which the algorithm selects patterns. Table 7 compares the results for the identification of sentences mentioning protest events on the development subset of the set of relevant documents. A pattern matches a sentence if for every word in the pattern, there is a token in the sentence that has the same lemma and lexical category, and the words from the pattern are matched at most 5 notional words apart. Additionally, we match some permutations of a pattern e.g. not only *organise protest* but also *protest organise*.

	top n patterns	precision	recall	F-score
bootstrap	500	0.54	0.03	0.06
bootstrap	2,000	0.64	0.13	0.21
bootstrap	5,000	0.49	0.23	0.31
baseline	500	0.37	0.27	0.31
baseline	2,000	0.32	0.39	**0.35**
baseline	5,000	0.28	0.44	0.34

Table 7: Performance of bootstrapped patterns for identification of sentences with event mentions. Rate of positive instances: 228/1,616.

Our results for bootstrapping should not be taken as conclusive: It is likely that a better choice of initial seeds and ranking metrics delivers superior bootstrapping patterns. On the other hand, we find that the straightforward baseline is difficult to beat and would recommend it for the protest domain over more complex, hard-to-tune bootstrapping methods. We use baseline patterns in our statistical anchor classifier.

7. Conclusion

We have presented the results of ongoing work on automating PEA. We have considered two parts of this complex problem: the identification of relevant news documents and the detection of protest event mentions in the text. We have applied some traditional NLP techniques to tackle these tasks and provided evaluation of their performance.

References

Steven Abney. *Semisupervised learning for computational linguistics*. CRC Press, 2007.

The ACE 2005 (ACE05) evaluation plan. `http://www.nist.gov/speech/tests/ace/ace05/doc/ace05-evalplan.v3.pdf`, 2005.

David Ahn. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8. Association for Computational Linguistics, 2006.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of Machine Learning Research*, 3:993–1022, 2003.

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454, 2006.

Mian Du and Roman Yangarber. Acquisition of domain-specific patterns for single document summarization and information extraction. In *The Second International Conference on Artificial Intelligence and Pattern Recognition (AIPR2015)*, page 30, 2015.

Susan Dumais, John Platt, David Heckerman, and Mehran Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the seventh international conference on Information and knowledge management*, pages 148–155. ACM, 1998.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

Alexander Hanna. Developing a system for the automated coding of protest event data. Available at SSRN: `http://ssrn.com/abstract=2425232` or `http://dx.doi.org/10.2139/ssrn.2425232`, 2014.

George Hripcsak and Adam S Rothschild. Agreement, the f-measure, and reliability in information retrieval. *Journal of the American Medical Informatics Association*, 12(3):296–298, 2005.

Ruihong Huang and Ellen Riloff. Multi-faceted event recognition with bootstrapped dictionaries. In *HLT-NAACL*, pages 41–51, 2013.

Swen Hutter. Protest event analysis and its offspring. In Donatella Della Porta, editor, *Methodological practices in social movement research*. Oxford University Press, 2014.

Dan Klein and Christopher D Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics, 2003.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

Kalev Leetaru. Automatic document categorization for highly nuanced topics in massive-scale document collections: The SPEED BIN program. `http://www.clinecenter.illinois.edu/publications/SPEED-BIN.pdf`, 2011.

Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, 2014.

Tara McIntosh and James R Curran. Weighted mutual exclusion bootstrapping for domain independent lexicon and template acquisition. In *Proceedings of the Australasian Language Technology Association Workshop*, volume 2008, 2008.

Tara McIntosh. *Reducing Semantic Drift in Biomedical Lexicon Bootstrapping*. PhD thesis, The University of Sydney, 2009.

George A Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

Robert Parker, Linguistic Data Consortium, et al. English Gigaword Fifth Edition LDC2011T07. 2011.

Jakub Piskorski and Roman Yangarber. Information extraction: Past, present and future. In *Multi-source, multilingual information extraction and summarization*, pages 23–49. Springer, 2013.

Martin F Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

Martin F Porter. Snowball: A language for stemming algorithms, 2001.

Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. Association for Computational Linguistics, 2009.

Ellen Riloff and Rosie Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI/IAAI*, pages 474–479, 1999.

Philip A Schrodt. Automated production of high-volume, real-time political event data. In *APSA 2010 Annual Meeting Paper*, 2010.

Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. BRAT: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107. Association for Computational Linguistics, 2012.

Bruno Wueest, Klaus Rothenhäusler, and Swen Hutter. Using computational linguistics to enhance protest event analysis. Available at SSRN: `http://ssrn.com/abstract=2286769` or `http://dx.doi.org/10.2139/ssrn.2286769`, 2013.

Roman Yangarber and Ralph Grishman. Customization of information extraction systems. In *Proceedings of International Workshop on Lexically-Driven Information Extraction*, pages 1–11, 1997.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.